

REMARKS

Claims 1-8, 10-17, and 19-26 are presented for consideration. Claims 9 and 18 were previously cancelled. No claim is currently amended. No matter is added.

Claims 1-26 stand rejected under U.S.C. §103(a) as being unpatentable over Ebner (U.S. Pat. 5,452,094), hereinafter Ebner. Claims 8,17 and 23 stand rejected under U.S.C. §103(a) as being unpatentable over Ebner in view of Uehara (U.S. Pat. 5,297,286).

A. Response to Arguments

In Part 2 of the Response to Arguments section of the current Office Action dated November 11, 2006, it is stated that,

"With regards to the modification of Star Micronics LogoStore disclosed by the Examiner in office action dated 6/28/2006, the applicant has not traversed the examiner's assertion of official notice. MPEP 2144.03 states 'a general allegation that the claims define a patentable invention without any reference to the examiner's assertion of official notice would be inadequate... If applicant does not traverse the examiner's assertion of official notice or applicant's traverse is not adequate, the examiner should clearly indicate in the next Office action that the common knowledge or well-known in the art statement is taken to be admitted prior art because the applicant either failed to traverse the examiner's assertion of official notice or that the traverse was inadequate.' Therefore the fact that self-installing programs as well as automatic detection of peripheral parameters were well known at the time of invention can now be regarded as prior art."

Applicants respectfully disagree.

Firstly, Applicants responded to the Office Action of June 28, 2006 by perfecting Applicant's priority date in a Response dated September 18, 2006, and thus invalidated the Star Micronics LogoStore reference and traversed any rejections based on the Star Micronics LogoStore, or in combination therewith. Applicants therefore reject the current Office Action's assertion that Applicant merely made, "a general allegation that the claims define a patentable invention without any reference to the examiner's assertion of official notice." By invalidating the Star Micronics LogoStore reference, Applicant did traverse any rejection base upon it, which include both Official Notices.

Secondly, by perfecting the foreign priority date of the present invention, the earliest priority date of the present application was effectively made earlier than the priority date upon which the previous Office Action had taken Official Notice.

Therefore, the previous Official Notice based on the unperfected priority date should no longer stand, and a new Official Notice would seem necessary.

Thirdly, a full reading of MPEP 2144.03 would maintain that a previous Official Notice might be asserted as prior art in the next Office Action, "if the rejection is to be maintained", but none of the rejections of the previous Office Action are maintained in the current Office Action since the Star Micronics LogoStore reference was invalidated in the previous Office Action Response.

In a telephone interview with Examiner Kang, it was agreed that it was improper for the Star Micronics LogoStore reference to be re-listed in the present Office Action, and that the previous rejections could not be maintained. Examiner Kang therefore agreed that it was improper for the previous Official Notices to be asserted as accepted prior art in the current Office Action. Examiner Kang further agreed to identify new prior art to support his view, if Applicants traverse the Official Notices in the current Office Action.

Applicants thus officially traverse both of the Office Action's Official Notices of fact asserting that both self-installing programs and automatic detection of peripheral parameters were well known at the time of the invention. Applicants traverse these two Official Notices for at least two reasons. Firstly, it is noted that neither the current, or any previous, Office Action has cited a prior art reference where the facts asserted to be well known are "capable of instant and unquestionable demonstration as being well-known", as specified by MPEP 2144.03. Secondly, Applicants note that after a diligent search for a reasonable time, Applicants were unable to identify dated support for either of the two Official Notice statements of fact. Thus, at least the MPEP 2144.03 requirement that any "Official notice unsupported by documentary evidence should only be taken ... where the facts asserted to be well-known ... are capable of instant and unquestionable demonstration as being well-known" is not met.

B. Claim Rejections

Claims 1-26 stand rejected under U.S.C. §103(a) as being unpatentable over Ebner (U.S. Pat. 5,452,094), hereinafter Ebner.

There appears to be a basic misunderstanding of the present invention since the cited Ebner reference appears to read most closely on what is explained in Applicant's prior art section, but is silent on all key features of Applicant's invention. Therefore, a brief summary of the prior art section is first presented, and thereby the problem being solved is highlighted. To facilitate later discussion of the cited prior art, the Ebner reference will also be addressed as it may highlight the prior art.

The Prior Art section of the present Application explains that since graphics, such as logo data, can be relatively large (especially those having color images), it is beneficial to store several logo data images/files within the printer attached to a point-of-sale device (i.e. POS system or cash register), and then to call up specific logo data by name for printing onto a receipt. Specifically, the Prior Art section states:

page 1, line 30 to page 2, line 1:

"Frequently printed logo data is therefore commonly stored in the non-volatile memory or storage of the POS terminal printer. The host system then only needs to send a specific command to the printer in order to print the logo. When the printer detects this command, it simply reads the logo specified by the print command from the non-volatile storage in order to print the logo."

continuing with page 2, lines 5-14:

"Modern POS printers are now capable of printing two or three colors. ... Color images are also significantly larger than black and white images, and processing color images accordingly requires more time. The need to quickly print color images on a POS printer therefore makes it even more important to store the image data as logo data in the printer. As the number and variety of printing applications for a POS printer increases, storing the logo data in the printer will be increasingly more important."

The above described solution is similar to that described by Ebner, who explains that higher end printers have ROM memory that store various logos, and these higher end printers have the ability to print a selected logo on a letter head area of a printed copy sheet as it is being printed (Ebner, col. 1, line 48 to col. 1, line 4). Ebner explains that the ROM memory chips have limited space, and cannot be easily updated with new logos. Specifically, Ebner explains that if one wishes to change the logos in the ROM chip, one needs to purchase a new ROM chip and have it custom burned with new logo data (Ebner, col. 2, lines 5-10). Presumably, this is because ROM chips are one-time-programmable. Ebner's solution is to replace the ROM chip with rewritable nonvolatile memory (for example EPROM, EEPROM, or Flash EPROM memory). In particular, because of the in-system-programmable solution described in Ebner's detailed description of his invention, he presumably proposes replacing the ROM chips with fully electrically rewritable memory, such as EEPROM or Flash EPROM memory. This touches upon the next problem described in the prior art section of the present application.

Having explained that it is customary to store various logo data in nonvolatile memory of POS system's printer, the Prior Art section of the present invention then addresses the problems associated with updating the logo in the nonvolatile memory of

the printer. As it is known in the art, some nonvolatile memory can be updated using a dedicated programmer, which requires direct access to the memory chip being updated. Newer nonvolatile memory chips support in-system reprogramming, but this requires the use of specific programming software. Thus, to use in-system programming, one needs to install a programming software utility into the POS host device attached to the printer, which requires updating each POS host device individually. Similarly, to use the dedicated programmer tool, one needs to detach the printer from the POS host device to access the printer's memory chip directly, which also requires updating each POS system individually. Specifically, the Prior Art section states:

Page 2, paragraphs 3-4

As described above, the effective printing speed (throughput) can be improved and the processing load on the host can be alleviated by storing the logo data in non-volatile storage in the printer. However, after the logo data is generated various additional steps are needed in order to store the logo data in the printer's non-volatile storage.

Consider a POS printer connected to the host device of the POS terminal at a checkout station in a store. To store the logo data in the POS printer while the printer remains connected to the POS terminal, a logo storage program must be installed in the host device of the POS terminal. For example, the logo data (image information) is saved as a file. The logo storage program installed in the host then reads this file and stores the logo data to the printer. This requires reading the file, and sending the logo data and a store command to the POS printer. These steps are also necessary when the logo data file is sent to the POS terminal via a LAN or other network and the logo data is transferred from the POS terminal to the printer.

Page 2, last paragraph

Installing such a logo storage program in each POS terminal is extremely time-consuming and cumbersome. It is alternatively possible to disconnect each POS printer from the POS terminal, and connect each printer to a dedicated logo data writer in order to store the logo data to the printer. This, however, requires that each printer be disconnected and then reconnected, which is even more time-consuming and further complicates the logo data storage operation.

As it would be understood, Ebner describes the process described in the 4th paragraph of page 2, posted above. Specifically with reference to Ebner's Fig. 3, Ebner describes logo data (image information) is saved as a file in a memory external to the printer (disk drive 27). A host device (i.e. PC 35) is installed with a logo storage program (i.e. programming software), and the host device PC 35 is in communication with nonvolatile memory 39 within the printer. In the present case, the host [35] then reads this [logo data] file and stores the logo data to the printer [29]. If Ebner's teachings were combined with the method described above in reference to the 4th paragraph of page 2 of the present Application, the combination might be interpreted to require, "reading the [logo data] file [from storage disk 27], and sending the logo data and a store command to the POS printer [35]." Thus, Ebner appears to follow the general flow chart of Fig. 9 of the present application, which describes a typical technology at the time of the present invention. A descriptions of the methods associated with Fig. 9 is found in page 10, lines 1-27 of the present Application. By contrast, the present invention follows more closely the flow chart of Fig. 2.

As is explained in the "Objects of the Invention" section of the present Application, a first objective of the present invention is to provide a print data (logo data) printer storage method that does not require prior installation of memory programming software on the POS host device, or POS terminal. A second object is to provide a system and method for creating a data storage file that when read by a POS host device, will provide all data files itself without need of reading any external library of data files and will automatically execute all necessary memory programming routines for programming the data files. Furthermore, the data storage file also implements any printer access routines necessary to cause the printer to provide access to its internal nonvolatile memory for programming.

Since none of these objectives are taught or suggested by Ebner, it would appear that the Office Action rejected several key features of the present invention based solely on previously made Official Notices. In a telephone interview with Examiner

Kang, it was noted that Office Action appears to equate the installing of new logo data in a printer with a self-installing program. Applicants explained that the data storage file of the present invention is executed in the POS host device, but no program is installed in the POS host device, or in any other device. Indeed, nothing is updated in the POS host device, itself. Rather, print data (not a program) is written to a memory store within a second device attached to the POS host device, i.e. to the printer, as is explained more fully below. Examiner Kang then explained that by "self-installing program", he meant any self-executing application, including one that might update information in a peripheral device. Examiner Kang suggested that an example of this might a self-executing program that updates font data in a printer. Nonetheless, it was agreed that additional prior art that more directly supports the Examiner's view should be identified.

In the same telephone interview, Applicants suggested that it is not clear how the Office Action's reference to "automatic detection of peripheral parameters (i.e., plug and play or PnP)" applies to the present invention. In a plug-and-play application, activation of a newly connected device results in the device announcing itself to the operation system, and periphery parameters (i.e. port number, interrupt level, etc.) are assigned by the operating system or by a user (see Uehara U.S. Pat. 5,297,286). In the present case, the printer is already installed and running. Furthermore, in an alternate embodiment of the present invention, a third device that is in communication with the POS device and with the printer interrogates the POS device and printer to ascertain their communication parameters (i.e. port connections) in order to create the data storage file that when sent to the POS device, will result in new logo data being written to the printer. A PNP application does not involve a third device external to a primary computing device and its connected periphery device. Examiner Kang stated that by "plug and play", he meant a first device that asks a second device how the second device wants data transmission to be organized. Examiner Kang further clarified that this is similar to how communication messages are packaged into transmission packets. Applicants responded that communication packets define different sections of a transmission for specific uses, such as for error correction, identifying the source of a communication, identifying the addressee of a communication, identifying the order of multiple related packets, the body of the data being sent, etc., but that the packet itself does not contain communication port information. This is because the physical link upon which the data packet is sent (i.e. Ethernet cable,

telephone wire, wireless, etc.) is supported by a different layer of the standard communication protocol, and not related to the data packet itself.

Examiner Kang further suggested that he was aware a device that, having two periphery devices, interrogates one device to determine its communication parameters in order to rely communication from second periphery device. Applicants inquired if perhaps the Examiner was referring to a printer and fax machine coupled to the same PC, but could not clearly identify the similarity and asked for further clarification. Examiner Kang clarified that by "plug-and-play", he meant any specific manner in which printer communication is organized. For example, he explained that some printers require that data be organized in specific ways, and this organization of data may be interpreted as "communication parameters". Applicants assume that the Examiner is referring to the use of some printers to organize print files to include a header section in which various print parameters are stored. However, applicants respectfully point out that such header information is typically related to the printer-driver, and generally does not contain communication parameters. Furthermore, Applicants respectfully remind the Examiner that claims cannot be read in vacuum, and must instead be read in light of the specification. Thus, attributing additional exotic meaning to "communication parameter", far beyond its meaning, as recited in the specification, might require additional explanation from the Examiner. Nonetheless, it was agreed that additional prior art that may more clearly support the Examiner's view should be identified.

It may be helpful to explain more clearly some of the key features of the present invention, which can be understood with reference to Fig. 1 of the present Application. Note that Fig. 1 does not show a printer nor a printer's host device. As it would be understood, however, the device of Fig. 1 (i.e. Data storage file compiler 4) is capable of communication with the printer, the printer's host device, or both. The structure of Fig. 1 may be used to create the data storage file of the present invention. Data storage file compiler 4 includes a logo generator (also referred to as a print data generator) 10, and can receives a source logo data file or image file that is generated by a logo editing tool or general purpose image editing tool. Data storage file compiler 4 further includes a logo storage 11 for holding created/received logo data, and a command set generator 12 for generating commands necessary for interfacing with a host device, printer, or both. Also included are a data storage file generator 18 and an output device 19. Data storage file compiler 4 produces a data storage file for each

specific type of host device and/or printer attached to the host device. In essence, the data storage file may include commands for execution by the host for communicating with an attached printer and sending processed logo data to the printer, and may further include programming commands to be executed by the printer for writing the processed logo data to its internal nonvolatile memory. In other words, it is not necessary for the printer or host device to process any image for printing, all processing is done beforehand, and prepackaged for writing directly to the printer's nonvolatile memory. A more detailed description can be found in Section C of the present Application beginning on page 11, line 26, where it is explained that:

"The data storage command set is a set of commands that can be executed by the printer 60 to store the logo data in the printer 60. The data transmission command set is a set of commands executable by the host 50 for sending the data storage command set to the printer. The data storage command set is therefore generated according to the system of commands that can be executed by the printer 60 in which the logo data is stored (that is, the printer command language), and the data transmission command set is generated according to the system of commands that can be executed by the host 50 to which the target printer is connected (that is, the host command language). ...

Section C further explains that transmission command set generator 13 can be used to interrogate a host device and/or printer in order to determine their communication ports, and write the appropriate communication ports into the data storage file that is sent to the host device. Alternatively, commands may be included so that the host device requests the printer's communication port information directly from the printer. More Specifically, Section C explains that,

"The transmission command set generator 13 of this embodiment, in turn, has a parameter input command set generator 15, a port detection command set generator 16, and a data transmission command set generator 17 for generating the data transmission command set to send the data storage command set and logo to the target printer. The parameter input command set generator 15 generates a command set that when executed by the host 50 receives input from the printer 60 indicating the printer port and communications parameters. The printer port and other communications conditions can be input when the host 50 runs this command. The port detection command set generator 16 generates a command set that when run by the host automatically detects the port to which the printer 60 is connected and the communications parameters.

Having provided a summary of the invention and the Ebner reference, one can now address other Office Action rejections. In reference to Claims 1, 10, and 20, the Office Action states that,

"Because Ebner makes no mention of direct memory access of the memory unit 25, it is necessary to send a "store" instruction to be executed by the copier processor, rather than directly accessing the memory and writing in the bits from the PC".

Applicants respectfully disagree and point to Fig. 3, where direct memory access of memory card 29 by PC 35 is clearly shown, in much the same way as direct memory access of hard drive 27 by PC 35 is shown. Ebner further appears to provide an option where memory card 29 directly programmed by PC 35 and then manually inserted into his printer. This further supports the appearance that Ebner does provide for direct memory access of memory card 29 by PC 35. In the above mentioned telephone interview, Examiner Kang explained that he mean to say that it is generally known in the art to use a "store" command for sending information to a periphery device's internal memory, although he conceded that such is not shown in Ebner.

The Office Action further states,

"Regarding step (c), because both the instruction for storing as well as the actual logo, in addition to any naming or descriptive text of the logo (col. 8, lines 25-32) are sent to the image forming apparatus from PC 35, this transmission comprises a "data storage file containing both the print data and the command data set."

Applicants respectfully disagree. It would appear that the term "file" is being misinterpreted. "The Authoritative Dictionary or IEEE Standard Terms", seventh edition, copyright 2000, defines a file as it relates to information transfer,

"file (3) (information transfer) One named collection of data."

Stated differently, it means that all the collect data (i.e. the print data and command data, in the present embodiment) must be collected under a single name, i.e. a single file name. In Ebner, storage commands, if any, are being sent as they are generated, not saved. Since no storage commands are saved, the commands cannot possibly be enclosed along with the print data within a file. Ebner makes no mention or suggestion of enclosing commands and print data within a single file. When Applicants brought this to the Examiner's attention, he explained that he meant to attribute this feature to the field of self-installing programs as being well known. Examiner Kang agreed to identify prior art to more fully support his position.

The Office Action then goes on to state,

"However, Ebner does not expressly disclose step (d), "a file output step for storing the data storage file in a data storage file to the host device via a communication path; wherein the print data is stored in non-volatile storage in the target printer in accordance

with the command data set upon the host device reading the data storage file."

Self-installing programs were well-known at the time of invention to those of normal skill in the art (official notice)."

As is stated above, Applicants respectfully traverse the Office Action's official notice, and request documentary support for such. Secondly, Applicants respectfully point out that in the present case, the host device is executing the commands, but nothing is being installed in the host device. Furthermore, no program of any kind is being installed in any device. Rather, a data set, i.e. print data is being written to a non-volatile memory located outside of the host device (i.e. located in a printer and written by the printer, not the host device). As it would be understood in the art, print data does not constitute a program.

The Office Action further equates a user executing a self-installing program with "upon the host device reading the data storage file". Applicants respectfully disagree since the host machine must have read the file when it first received it. By contrast, user execution of a self-installing program would require the user selecting a program file that is pre-existing on the host device, and must therefore have been previously read and stored.

In regards to claims 3 and 12, the Office Action appears to be confusing high level language (as written by a human operator) with machine language. Machines, in general do not communicate in "high level instructions", but rather communicate in machine code. Computer programs must therefore be compiled into machine code prior to execution by a machine. Thus, it is incorrect to assume that a machine would see the command "store" in English, within an executing program, and then translate it into machine code for execution by a printer. This operation would likely occur when the written program is first compiled for use in a machine. Nonetheless, this does not apply to the present case since, as shown in Fig. 1, the logo storage file of the present invention is written in by a machine, and is thus already in machine code. English terms equivalent to the low level 0's and 1's are used in the discussion in order to facilitate explanation of the present invention. However, it is self evident that when the structure of Fig. 1 generates code executable by a machine, it is most likely already in machine code, not in a high level (i.e. human) language.

In reference to claims 7, 16, and 24, the Office Action asserts that,

"Ebner does not expressly disclose "an executable command set which, when run by the host device, detects the communication

parameters, and sends the data storage command set and print data to the target printer according to the detected communication parameters."

Automatic detection of peripheral parameters (i.e., plug and play or PnP) was a well-known concept and in widely used in the industry at the time of invention (official notice)."

Firstly as is explained above, Applicants respectfully traverse the Office Action's official notice that automatic detection of peripheral parameters were well known at the time of the present invention. Secondly, even if one were to apply "plug and play" techniques to the teachings of Ebner, one would still not achieve the present invention. In a plug-and-play application, activation of a newly connected device results in the device announcing itself to the operation system, and periphery parameters (i.e. port number, interrupt level, etc.) are assigned by the operating system or by a user (see Uehara U.S. Pat. 5,297,286). In the present case, the printer is already installed and running.

Lastly, Applicants respectfully point out that the structure defined by the family of claims depending from claim 20 recite an apparatus that can communicate with a host device and/or with a printer attached to the host device, and can separately interrogate both devices to determine their communication criteria (i.e. language, port information, capabilities, etc.). The recited apparatus generates a logo storage file based on the information such that when executed by the host device will control the printer to update its nonvolatile memory with print data encapsulated within the logo stage file.

A reason why this feature is useful is that the present invention is particularly directed to POS systems, i.e. cash registers. Such systems routinely have many POS stations connected by a network to a central controller machine, which keeps track of sales transactions, price changes, etc. Since it is likely that a particular store may have several different POS host device and printer combinations, it is beneficial for the central controller to be able to interrogate each POS system/station and determine its capabilities in order to generate a custom logo storage file for it. Applicants are at a loss to determine how Ebner, or any cited prior art, teach or suggest this structure. Ebner describes no other device besides his PC 35, external disk drive 27, and copier 10. Applicants respectfully request identification, in the cited prior art, of an apparatus in communication with a host device and with the host device's attached printer, that operates to generate a logo storage file capable of

causing the host device to extract logo data within the logo storage file and transfer the extracted logo data to the printer.

In view of the foregoing amendments and remarks, Applicants respectfully request favorable reconsideration of the present application.

Respectfully submitted,

/Rosalio Haro/

Rosalio Haro

Registration No. 42,633

Please address all correspondence to:

Epson Research and Development, Inc.
Intellectual Property Department
2580 Orchard Parkway, Suite 225
San Jose, CA 95131
Phone: (408) 952-6131
Facsimile: (408) 954-9058
Customer No. 20178
Date: February 27, 2007